

Thing Directory: Simple and lightweight registry of IoT device metadata

Farshid Tavakolizadeh¹ and Shreekantha Devasya¹

¹ Fraunhofer Institute for Applied Information Technology, Sankt Augustin, Germany

DOI: [10.21105/joss.03075](https://doi.org/10.21105/joss.03075)

Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Editor: [George K. Thiruvathukal](#) ↗

Reviewers:

- [@yadudoc](#)
- [@sarathsp06](#)

Submitted: 29 December 2020

Published: 14 April 2021

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Statement of Need

The fast emergence of IoT (Internet of Things) technologies has influenced scientific communities to embrace novel sources of information and their potential use in various domains. While the vast amount of sensory data is beneficial, the lack of uniform access interfaces hinders researchers from efficient exploitation. A structured, yet flexible registry is needed to model device metadata and allow exploration and interaction with such devices. In the IoT context, the Things are physical devices (e.g., sensors, actuators, gateways) or virtual ones (e.g., digital twins, proxies, aggregated data sources). An ideal metadata registry for such Things should have a flat learning curve and be easily deployable. This would allow researchers to focus less on interoperability and more on fast prototyping and data application. Registries that are based on established standards are preferred since they can incorporate metadata with many existing Things out-of-the-box. Moreover, the registry software should be lightweight, easily deployable, and free of rigid requirements to support fast prototyping across a wide range of use cases from the edge to the cloud.

Thing Directory

Thing Directory is a searchable registry of metadata for Things. The API is based on W3C Web of Things (WoT) Discovery ([Cimmino et al., 2020](#)), a specification for secure discovery of Things. The registry uses JSON-LD (JSON for Linked Data) encoding by default. The JSON format is human-readable and portable; the linked data support makes the data machine-interpretable. The architecture of the registry is modular with a pluggable storage backend, allowing connection to various database systems using drivers. The current implementation, backed by an embedded LevelDB storage, can be deployed on highly constrained single-board computers such as the Raspberry Pi Zero series. It has very minimal idle processing and memory footprints and can scale on demand to utilize all locally available resources. More powerful storage backends can be added to create a horizontally scalable directory in cloud environments.

The application is packaged as binary distributions, Debian packages, as well as Docker images for easy deployment on a variety of platforms. The data model of the metadata is based on W3C WoT Thing Description (TD) ([Kaebisch et al., 2020](#)) which is extensible by design. Thing Directory validates all inputs using a JSON-Schema, describing the data model. This makes it possible to extend the server's structured data model and validation mechanism with no programming. The various modules of the system are provided as re-usable Go libraries which can be imported by other applications to build functionalities around Thing Descriptions.

Use case: Assessment of Building Energy Efficiency

Construction companies often deal with the challenge of delivering target energy-efficient buildings, given specific budget and time constraints. Energy efficiency, as one of the key factors for renovation investments, depends on the availability of various data sources to support the renovation design and planning. These include climate data and building material along with residential comfort and energy consumption patterns.

As part of the pre-renovation activities, the construction planners deploy various sensors to collect relevant data over a period. Such sensors become part of a wireless sensor network (WSN) and expose data endpoints with the help of one or more gateways. Depending on the protocols, the endpoints require different interaction flows to securely deliver current and historical measurements. The renovation applications need to discover the sensors, their endpoints, and how to interact with them based on search criteria such as the physical location, mapping to the building model, or measurement type.

The Thing Directory supports engineers in the assessment of building energy efficiency by providing the means to collect and discover the metadata of deployed sensors in an easy and standardized way. Instances of Thing Directory have been deployed in four renovation sites (two apartments, two buildings) across Europe as registries of wireless sensors which are locally accessible over Z-Wave or WiFi. The API has been integrated into applications for profiling of resident usage of building systems, building information modeling, and process modeling and automation. Such applications query sensor metadata based on zoning and sensor types. Once discovered, the metadata provides these applications with necessary details on how to authenticate and query data. Since the Thing Directory is based on an open standard, being integrated with it adds interoperability with WSNs beyond the scope of this use case. The applications will be able to interact with the growing number of compliant WoT devices.

Related Work

There are multiple attempts to modeling and discovering the connected Things and their interfaces. OGC SensorThings API ([Liang et al., 2016, 2017](#)) has been a successful model for the representation of Things. Sensorthings API consists of two parts: sensing and tasking. The popular implementations of the OGC SensorThings are FROST ([van der Schaaf & Jacoby, 2016](#)) and GOST ([Geodan, 2016](#)) which support mainly the sensing part. FROST has preliminary tasking support. These solutions focus on centralized deployments and are not suitable for a federated scenario and gateways with limited computational power. On top of that, the specification which they are based upon, enforces both metadata and observation modeling. That is not practical in IoT scenarios with heterogeneous data formats and interfaces. The survey articles by [Bröring et al. \(2016\)](#) and [Khalil et al. \(2020\)](#) discuss and categorize several other technologies related to discovery in the IoT field and evaluate them.

The WoT TD ([Kaebisch et al., 2020](#)) covers a wide range of Things by providing the semantics to describe the textual metadata, interaction affordances, data schemas, and relations. Thingweb Directory ([Thingweb, 2016](#)) has implemented the discovery of WoT TDs using a proprietary API which does not comply with W3C WoT Discovery ([Cimmino et al., 2020](#)) standard. The Thing Directory complies with both W3C WoT Discovery and W3C WoT TD.

Acknowledgement

This work was conducted as part of the BIMERR project, a European Commission's Horizon 2020 research and innovation program under grant agreement No 820621. The resulting software is released as part of LinkSmart, an open-source IoT prototyping platform.

References

- Bröring, A., Datta, S. K., & Bonnet, C. (2016). A Categorization of Discovery Technologies for the Internet of Things. *ACM International Conference Proceeding Series, 07-09-Nov*, 131–139. <https://doi.org/10.1145/2991561.2991570>
- Cimmino, A., McCool, M., Tavakolizadeh, F., & Toumura, K. (2020). *Web of things (WoT) discovery* [W3C Working Draft]. W3C. <https://www.w3.org/TR/wot-discovery/>
- Geodan. (2016). GOST (go-SensorThings). In *GitHub repository*. GitHub. <https://github.com/gost/server>
- Kaebisch, S., Kamiya, T., McCool, M., Charpenay, V., & Kovatsch, M. (2020). *Web of things (WoT) thing description 1.1* [W3C Working Draft]. W3C. <https://www.w3.org/TR/wot-thing-description/>
- Khalil, K., Elgazzar, K., Seliem, M., & Bayoumi, M. (2020). Resource discovery techniques in the internet of things: A review. *Internet of Things, 12*, 100293. <https://doi.org/10.1016/j.iot.2020.100293>
- Liang, S., Huang, C.-Y., & Khalafbeigi, T. (2016). *OGC SensorThings API part 1: Sensing, version 1.0*.
- Liang, S., Khalafbeigi, T., & Luo, K. (2017). *OGC SensorThings API part 2 – tasking core*.
- Thingweb. (2016). Thingweb-directory. In *GitHub repository*. GitHub. <https://github.com/thingweb/thingweb-directory>
- van der Schaaf, H., & Jacoby, M. (2016). FROST server. In *GitHub repository*. GitHub. <https://github.com/FraunhoferIOSB/FROST-Server>